ST-LLM: Large Language Models Are Effective Temporal Learners

Ruyang Liu[*1,3] Chen Li[*2] Haoran Tang[1,3] Yixiao Ge[2] Ying Shan[2] Ge Li ✉[1]

[1] School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University {ruyang@stu,hrtang@,geli@ece}.pku.edu.cn
[2] ARC Lab, Tencent PCG [3] Peng Cheng Laboratory
{palchenli,yixiaoge,yingsshan}@tencent.com

Abstract. Large Language Models (LLMs) have showcased impressive capabilities in text comprehension and generation, prompting research efforts towards video LLMs to facilitate human-AI interaction at the video level. However, how to effectively encode and understand videos in video-based dialogue systems remains to be solved. In this paper, we investigate a straightforward yet unexplored question: Can we feed all spatial-temporal tokens into the LLM, thus delegating the task of video sequence modeling to the LLMs? Surprisingly, this simple approach yields significant improvements in video understanding. Based upon this, we propose ST-LLM, an effective video-LLM baseline with **S**patial-**T**emporal sequence modeling inside **LLM**. Furthermore, to address the overhead and stability issues introduced by uncompressed video tokens within LLMs, we develop a dynamic masking strategy with tailor-made training objectives. For particularly long videos, we have also designed a global-local input module to balance efficiency and effectiveness. Consequently, we harness LLM for proficient spatial-temporal modeling, while upholding efficiency and stability. Extensive experimental results attest to the effectiveness of our method. Through a more concise model and training pipeline, ST-LLM establishes a new state-of-the-art result on VideoChatGPT-Bench and MVBench. Codes have been available at https://github.com/TencentARC/ST-LLM.

Keywords: ST-LLM · Dynamic Masking · Global-Local Input

1 Introduction

Large Language Models (LLMs), such as GPT [1,9], PaLM [3,16] and LLaMA [43,44], have achieved remarkable success owing to their formidable language understanding and generation abilities, signaling a promising advancement towards artificial general intelligence. Driven by the widespread adoption of LLMs, research on Large Visual-Language Models (LVLMs) [14,29,55] has emerged to extend the capabilities of LLMs to process visual signals, which have revealed proficiency in image-based conversations. However, in contrast to image inputs, videos feature heavier input and additional temporal information. Developing a

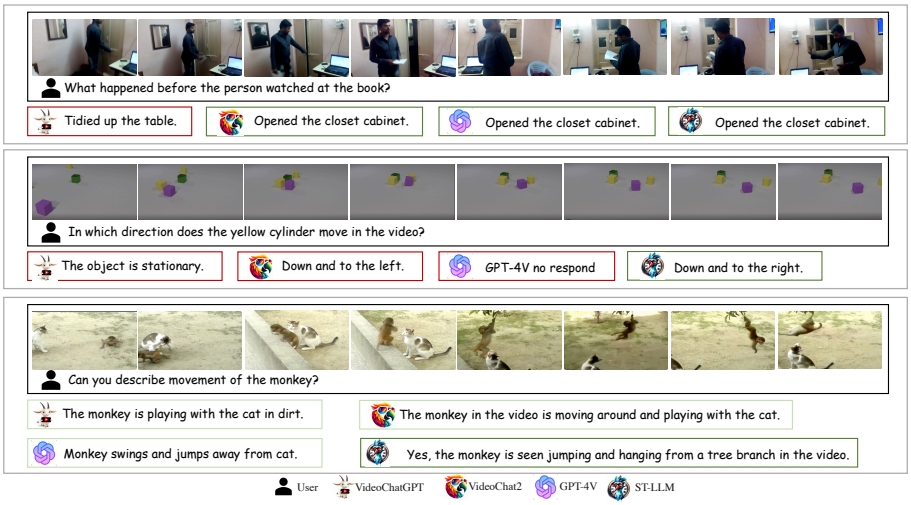


Fig. 1: Qualitative comparisons among top-performance video LLMs. We illustrate two cases from MVBench [25] (above) and one case from a YouTube video (below).

large video-language model capable of effectively extracting meaningful spatial-temporal information from intricate video signals presents a significant challenge.

Recently, several preliminary attempts have surfaced aimed at extending LLMs to facilitate video conversation [19, 20, 24, 25, 27, 31–33, 53], which unlock the capability of generating generic summaries of video content. Despite these efforts, existing video LLMs continue to exhibit shortcomings in achieving satisfactory performance in video comprehension, particularly in the realm of understanding content that relies heavily on temporal dynamics. As depicted in Fig. 1, contemporary top-performing video LLMs demonstrate robust understanding capabilities for actions reliant on static contexts. However, their proficiency in comprehending scenes involving motion remains constrained. They encounter challenges in discerning even the most basic direction of object movement, let alone the complex motion or scene transition.

This issue may stem from the inherent difficulty of temporal modeling within conversational systems. Historically, achieving robust video encoding has often required substantially more time and memory resources than images, which is impractical based on LLMs. As illustrated by Fig. 2(a), early video LLMs tend to adopt mean pooling on temporal dimension [24, 32, 33, 53], a method that, while efficient, is inadequate for handling dynamic temporal sequences. Therefore, recent models frequently integrate additional structures for temporal sampling and modeling [19, 20, 25, 27]. While these methods offer improved effectiveness compared to average pooling, they entail increased storage and demand extensive GPU time for training from scratch, often involving two or even three-stage pretraining to align new modules.

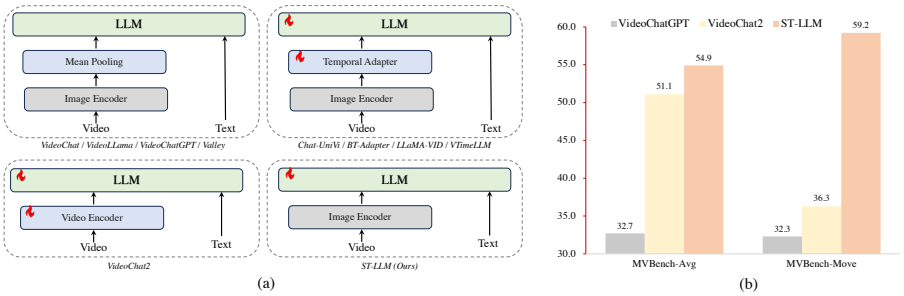


Fig. 2: (a) Comparison among different video LLMs based on the way of inputting visual tokens. (b) Quantitative comparison of representative video LLMs, presenting the average results of all 20 metrics and temporal-sensitive motion-related metrics from MVBench, which encompass Moving Direction, Moving Count, and Moving Attribute.

Inspired by the recent success of joint spatial-temporal-text modeling in visual generation [8,36], in this work, we investigate a simple yet unexplored idea: Given the robust sequence modeling capabilities inherent to LLMs, what if we input all visual tokens into the LLM and delegate the task of modeling spatial-temporal sequences to the LLM itself? Meanwhile, there are two additional challenges: (1) The inclusion of all visual tokens significantly increases the context length within LLMs, particularly for long videos, rendering the processing of a large number of frames unaffordable. (2) LLM may struggle to handle videos of varying lengths, potentially leading to hallucinations when there is a discrepancy between the number of frames in testing and training.

To address these issues, we propose ST-LLM, a simple but powerful baseline of video LLM. As depicted in Fig. 2(a), with raw spatial-temporal tokens inside LLM, we harness the LLMs' robust sequence modeling capabilities for effective temporal modeling. Meanwhile, we introduce a novel dynamic video token masking strategy along with masking video modeling during training. With this approach, we reduce the length of sequences input to the LLM while significantly improving the robustness of videos of varying lengths during inference. To accommodate particularly long videos, we have devised a unique global-local input mechanism. This involves employing mean pooling of a large number of frames to generate residual input for a smaller subset of frames. Through this asymmetric design, we can process input from a large number of video frames while preserving the operations of modeling video tokens within LLMs.

As a result, we maintain input tokens of comparable length to most video LLMs while effectively harnessing the LLM's capacity to comprehend videos. Furthermore, as ST-LLM does not introduce additional modules, it does not necessitate expensive alignment pre-training. This enables ST-LLM to directly leverage existing state-of-the-art image conversational models, resulting in significantly reduced GPU time compared to other state-of-the-art video LLMs. Extensive experiments are conducted to verify the effectiveness of ST-LLM. Qualitatively, as depicted in Fig. 1, ST-LLM demonstrates a superior ability to

understand dynamics compared to other video LLMs. Quantitatively, ST-LLM achieves new state-of-the-art performance across various contemporary video benchmarks, including MVBench [25], VideoChatGPT-Bench [33] and zero-shot QA-Evaluation. Specifically, as illustrated in Fig. 2(b), the superiority of ST-LLM is particularly evident in metrics related to the temporal-sensitive motion, which underscores the expertise of our model in temporal understanding.

2 Related Works

LLMs and Image LLMs. Recent years have witnessed remarkable advancements in the evolution of Large Language Models (LLMs). Inspired by InstructGPT [35] and the widely recognized commercial model ChatGPT [34], the academic community has seen a proliferation of open-source LLMs, such as LLaMA [43], Alpaca [41], Vicuna [13], and LLaMA 2 [44], which have become foundational components for a myriad of research endeavors. Meanwhile, LLMs, being large-scale transformer models, inherently possess strong sequence modeling capabilities. This inspired us to explore the feasibility of entrusting the task of spatial-temporal sequence modeling to LLMs. The success of LLMs has spurred increasing interest in the development of Multimodal Large Language Models (MLLMs). Notable breakthroughs include works such as Flamingo [2], BLIP2 [23], and PaLM-E [16], which have successfully bridged the gap between vision models and LLMs. Inspired by the concept of instruction tuning for LLMs, a series of works [12, 14, 29, 55] leveraged open-source chatbots for visual instruction tuning, thus facilitating support for image-based conversations. It is noteworthy that in the pre-LLM era, video models were commonly constructed based on pre-trained 2D image models [7, 11, 30, 45]. This may be attributed to the significant computational demands of video processing, combined with the relatively limited scale and diversity of video data compared to images. Therefore, we believe that a well-pretrained image dialogue model could similarly offer training cost and performance benefits to video LLMs.

Video LLMs. The emergence of MLLMs quickly extended into the domain of video as well [19, 20, 24, 25, 27, 31–33, 53]. Early models like VideoChat [24], VideoChatGPT [33] and Valley [32] generate video instruction tuning data through GPT to enable video conversations. At the model level, these approaches typically involve the use of mean pooling to aggregate the encoding results of individual frames before feeding them into the LLMs. While efficient, it is evident that this method is inadequate for effective temporal modeling. Hence, some subsequent models have initiated exploration into adaptations of image models to video-specific requirements. For instance, BT-Adapter [31] proposed a lightweight adapter to extend the capabilities of image LLMs. Chat-UniVi [20] introduces DPC-KNN to cluster dynamic visual tokens. Moreover, VideoChat2 [25] deviated from using CLIP [38] and directly introduced a dedicated video encoder. Although these methods offer improvements over mean pooling, the inclusion of additional modules typically necessitates intensive training. For example, in the case of VideoChat2, aligning the newly introduced video encoder entails

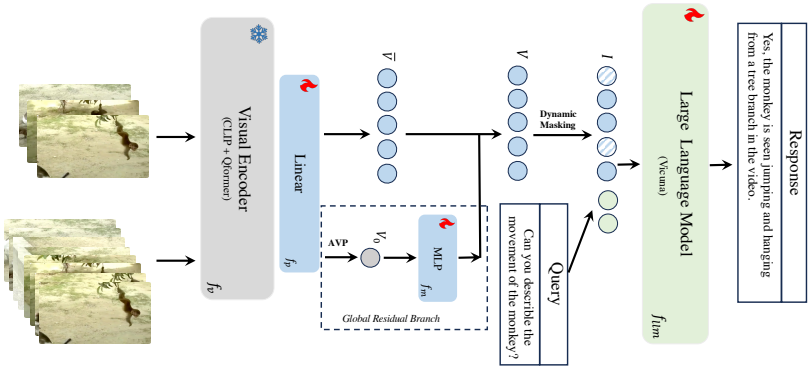


Fig. 3: The overview of ST-LLM for generating responses on the input video and instructions. ST-LLM directly feeds the spatial-temporal sequence into the LLM and employs dynamic masking and global-local input for efficiency and robustness.

a three-stage training involving 30 million visual-language samples. In contrast, ST-LLM adopts a more direct approach by entrusting the task of visual sequence modeling to LLMs, which is characterized by its conciseness, reduced training requirements, and, according to our experimental findings, superior effectiveness.

Joint Spatial-Temporal Modeling. Prior to the emergence of LLMs, joint spatial-temporal modeling had demonstrated effectiveness in both video-only and video-language pretraining [26,42,49]. More recently, inspired by DIT [36], the popular video generation model Sora [8] has shown success in simultaneously processing video tokens and text using the transformer. Despite its potential, joint spatial-temporal-text modeling remains relatively uncommon in video LLMs. While VideoChat2 [25] employed a joint-ST video encoder, it necessitated extensive pretraining. A closer relative, InstructBLIP [14] incorporated joint spatial-temporal sequences into LLM for zero-shot evaluation on video datasets. However, this approach led to significant hallucinations due to the absence of video instruction tuning. To the best of our knowledge, ST-LLM is the first open-source video LLM to adopt joint spatial-temporal-text modeling. Moreover, ST-LLM introduces a series of design innovations to mitigate the limitations associated with incorporating all video tokens within LLM.

3 Methodology

In this section, we elaborate on the structure and training methodology of ST-LLM. Firstly, we present a description of the model architecture, which incorporates all spatial-temporal tokens within the LLM, in Section 3.1. Subsequently, we delve into the masking mechanism and training objectives in Section 3.2. Finally, we introduce the global-local input approach in Section 3.3. The overarching framework of our method is illustrated in Fig. 3.

3.1 Video Tokens Inside LLM

As depicted in Fig. 3, visually, ST-LLM closely resembles an image chatbot, consisting of a visual encoder, a linear projection, and an LLM. Instead of opting for the commonly used CLIP-L/14 [38], we have selected BLIP-2 [23] as our visual encoder, whose Q-Former efficiently compresses redundant image tokens into fewer visual tokens. To facilitate the input of raw videos for the image encoder, we adopt the strategy of treating each frame as an individual image. Given a video v with T frames, we assume that the image encoder f_{v} encodes each frame into K tokens. Hence, we obtain all visual tokens V to be fed into the LLM as follows:

$$V = f_{\mathrm{p}}(f_{\mathrm{v}}(v)), \text{ where } V = \{V_i\}_{i=1}^{T}, \ V_i = \{v_{(i,j)}\}_{j=1}^{K}, \tag{1}$$

where f_{p} is the visual projection layer. Then, we concatenate all visual tokens to form a joint spatial-temporal [7] visual sequence $V \in \mathbb{R}^{(T*K)\times D}$, where D is the number of embedding dimensions. After being passed through the word embedding layers, the text tokens $C = \{c_i\}_{i=1}^{N}$ are directly concatenated with all spatial-temporal tokens, to organize the input I to the LLM as follows:

$$I = [V : C] = \{v_{(1,1)}, v_{(1,2)}, ..., v_{(1,K)}, v_{(2,1)}, ..., v_{(T,K)}, c_1, c_2, ..., c_N\}. \tag{2}$$

Indeed, several alternative strategies exist for organizing the input tokens before feeding them into the LLM. These include: (1) Adding separator tokens between frame-wise tokens or visual-text tokens, to demarcate the boundaries between modalities or frames. (2) Adding spatial-temporal position embeddings specifically for visual tokens, to facilitate better spatial-temporal relationship understandings. However, our experiments indicate that simplicity often yields the best. Adding numerous separator tokens introduces additional overhead without necessarily improving performance. Moreover, LLMs come with Rotary Position Embeddings [39] that are already effective in distinguishing the positions of all spatial-temporal tokens and text tokens.

3.2 Training with Dynamic Masking

With video tokens modeling inside LLM, we can leverage the potent sequence modeling capabilities inherent in LLMs to understand the information encapsulated within spatial-temporal sequences. However, integrating all video tokens into LLM significantly increases the context length, imposing a considerable computational burden and rendering it unfeasible to input additional frames for lengthy videos. Meanwhile, experimental findings suggest that during testing, utilizing uncompressed tokens is more sensitive to variations in the number of frames than mean pooling. Consequently, when a substantial dissonance exists between the number of frames in training and testing, a noticeable degradation in performance ensues.

To tackle the aforementioned issues, we first propose masking the visual tokens during training. Masking modeling has achieved significant success in natural language processing [15] and video-language pretraining [26,31,42]. Although

masking is rarely employed in autoregressive LLMs, given our aim to leverage LLMs for encoding spatial-temporal sequences, we can still utilize masking modeling. Specifically, we keep the text tokens in I unchanged, while applying a mask to the video tokens. This mask randomly masks ρ of all video tokens, irrespective of their position across different frames. Different from previous works [26,31], we adopt a dynamic masking strategy, where the masking rate is randomly sampled from a normal distribution as follows:

$$\rho \sim \mathbb{N}(0.5, \sigma), \quad 0.3 \leq \rho \leq 0.7, \tag{3}$$

where σ represents the variance of the normal distribution. This strategy ensures that the length of the spatial-temporal sequence varies continuously while maintaining a consistently high average masking rate of 50%. As a result, it minimizes training costs and notably enhances robustness during inference.

Furthermore, building upon dynamic masking, we have formulated the Masked Video Modeling (MVM) objective to encourage the LLM to grasp spatial-temporal dependencies. Unlike the expensive masked token recovery [6,37], we have opted for unmasked token reconstruction. Specifically, besides the masked sequence I, we conduct an extra non-grad forward pass for the unmasked sequence $\hat{I} = [\hat{V} : C]$, serving as the reference output. Then, we select unmasked tokens from $f_{llm}(I)$ and the corresponding tokens from $f_{llm}(\hat{I})$ based on their positions, computing the Mean Squared Error (MSE) between the selected pairs as follows:

$$\mathcal{L}_{mvm} = \frac{1}{(1-\rho)KT} \sum_{i=1}^{T} \sum_{j=1}^{K} (v_{i,j}^{-1} - \hat{v}_{i,j}^{-1})^2, \quad (i,j) \notin M, \tag{4}$$

where M denotes the set of masked tokens' indices, $v_{i,j}^{-1}$ is from $f_{llm}(I)$, and $\mathcal{L}_{mvm}$ represents our MVM objective. Finally, our overall loss $\mathcal{L}$ is composed of $\mathcal{L}_{mvm}$ and the LLM decoder loss $\mathcal{L}_{llm}$, yielding $\mathcal{L} = \mathcal{L}_{mvm} + \mathcal{L}_{llm}$. By integrating these two loss components, we can encourage the LLM to effectively respond to questions derived from the video content while improving its ability to model temporal and spatial dependencies.

3.3 Global-Local Input

Although dynamic masking partially mitigates the challenge of processing lengthy input sequences, it fails to fully resolve the issue for extremely long videos that entail inputting numerous frames, rendering the context length still impractical to handle. Therefore, we have devised an additional module to tackle the issue of excessively long videos. To elaborate, given a lengthy video with a large T, we still commence by encoding each frame individually to derive V. Subsequently, we proceed to derive the global video representation V_0 through average pooling on the frame-wise tokens:

$$v_{(0,j)} = \frac{1}{T} \sum_{i=1}^{T} v_{(i,j)}, \quad V_0 = \{v_{(0,j)}\}_{j=1}^{K}. \tag{5}$$

Next, we average-sample $\overline{T}$ frames from the T total frames. All tokens from these $\overline{T}$ frames are concatenated to produce a joint spatial-temporal sequence $\overline{V} \in \mathbb{R}^{(\overline{T}*K)\times D}$, which serves as the local video representation. Finally, the global-local input for the LLM, denoted as $\overline{I}$, is constructed as follows:

$$\overline{I} = [\overline{V} + f_m(V_0) : C], \tag{6}$$

where f_m is a simple MLP projector with upsampling projection initialized with zeros. With this global-local input design, the low fps spatial-temporal sequences within the LLM can gradually incorporate information from the high fps branch. This approach allows the model to benefit from the LLM's ability to model temporal sequences within a limited context while also considering the global information of long videos.

4 Experiments

In this section, we have performed comprehensive experimental evaluations of ST-LLM, covering crucial settings, comparisons, and ablation studies. For a more detailed account of experimental settings, ablation studies, visualizations, and limitations analysis, please consult the appendix.

4.1 Experiment Setup

Benchmarks. To evaluate the video understanding capabilities of ST-LLM, we primarily conduct assessments on three benchmarks: MVBench [25], VideoChat-GPT Bench [33], and zero-shot video QA benchmark. MVBench comprises 20 challenging video tasks, each consisting of 200 samples in the form of multiple-choice questions. These tasks provide a comprehensive and objective assessment of a model's ability to understand videos. VideoChatGPT-Bench gathers videos from ActivityNet [10] and uses GPT to evaluate the quality of video conversations across five dimensions. Additionally, the zero-shot video QA benchmark entails GPT-based assessments of various open-source video QA datasets.

Implementation Details. In our study, we adopt state-of-the-art image dialogue models as the baseline and proceed directly with video instruction tuning. Unless otherwise specified, our model is initialized with InstructBLIP [14] in all experiments, whose LLM is Vicuna-v1.1 [13] with 7B parameters. Additionally, we employ Minigpt4-v1 [55] as a basic model to assess the effectiveness across various image models in Table 6. Following previous research [20, 27, 28], we adopt full-finetuning on the LLM to ensure a fair comparison. The results of LoRA are also included as ablation results in Table 6. σ is set as 0.1, ensuring that over 95% of the mask rate falls within the range of 0.3 to 0.7. During training, we sample 16 frames per video. Considering dynamic masking, an average of 256 tokens per video are input into the LLM, which is the same as LLaVA [29]. For all benchmarks during inference, frames are sampled at a frame rate of 1, and the number of local frames is kept between 4 and 16. This strategy closely

Table 1: Comparisons on MVBench. Except for BLIP2 and Otter, all models are built upon LLaMA-1-7B for fair comparisons. "Avg" denotes the average of all 20 metrics. The leaderboards for each task can be found in the Appendix. The compared methods include: (1)Rd: random guesses; (2) mOI: mPLUG-Owl-I [51]; (3) LMA: LLaMA-Adapter [54] (4)B2: BLIP-2 [23]; (5)OI: Otter-I [22]; (6)MG: MiniGPT-4 [55]; (7)IB: InstructBLIP [14]; (8)LV: LLaVA [29]; (9)OV: Otter-V [22]; (10)mOV: mPLUG-Owl-V [51]; (11)VCG: VideoChatGPT [33]; (12)VLM: VideoLLaMA [53]; (13)VC: VideoChat [24]; (14)VC2: VideoChat2 [25].

Metric	Rd	mOI	LMA	B2	OI	MG	IB	LV	OV	mOV	VCG	VLM	VC	VC2	Ours
AS	25.0	25.0	23.0	24.5	34.5	16.0	20.0	28.0	23.0	22.0	23.5	27.5	33.5	**66.0**	**66.0**
AP	25.0	20.0	28.0	29.0	32.0	18.0	16.5	39.5	23.0	28.0	26.0	25.5	26.5	47.5	**53.5**
AA	33.3	44.5	51.0	33.5	39.5	26.0	46.0	63.0	27.5	34.0	62.0	51.0	50.0	83.5	**84.0**
FA	25.0	27.0	30.0	17.0	30.5	21.5	24.5	30.5	27.0	29.0	22.5	29.0	33.5	**49.5**	44.0
UA	25.0	23.5	33.0	42.0	38.5	16.0	46.0	39.0	29.5	29.0	26.5	39.0	40.5	**60.0**	58.5
OE	33.3	36.0	53.5	51.5	48.5	29.5	51.0	53.0	53.0	40.5	54.0	48.0	53.0	58.0	**80.5**
OI	25.0	24.0	32.5	26.0	44.0	25.5	26.0	41.0	28.0	27.0	28.0	40.5	40.5	71.5	**73.5**
OS	33.3	34.0	33.5	31.0	29.5	13.0	37.5	41.5	33.0	31.5	40.0	38.0	30.0	**42.5**	38.5
MD	25.0	23.0	25.5	25.5	19.0	11.5	22.0	23.0	24.5	27.0	23.0	22.5	25.5	23.0	**42.5**
AL	25.0	24.0	21.5	26.0	25.5	12.0	23.0	20.5	23.5	23.0	20.0	22.5	27.0	23.0	**31.0**
ST	25.0	34.5	30.5	32.5	55.0	9.5	46.5	45.0	27.5	29.0	31.0	43.0	48.5	**88.5**	86.5
AC	33.3	34.5	29.0	25.5	20.0	32.5	**42.5**	34.0	26.0	31.5	30.5	34.0	35.0	39.0	36.5
MC	25.0	22.0	22.5	30.0	32.5	15.5	26.5	20.5	28.5	27.0	25.5	22.5	20.5	42.0	**56.5**
MA	33.3	31.5	41.5	40.0	28.5	8.0	40.5	38.5	18.0	40.0	39.5	32.5	42.5	58.5	**78.5**
SC	33.3	40.0	39.5	42.0	39.0	34.0	32.0	47.0	38.5	44.0	**48.5**	45.5	46.0	44.0	43.0
FP	25.0	24.0	25.0	27.0	28.0	26.0	25.5	25.0	22.0	24.0	29.0	32.5	26.5	**49.0**	44.5
CO	33.3	37.0	31.5	30.0	27.0	29.5	30.5	36.0	22.0	31.0	33.0	40.0	41.0	36.5	**46.5**
EN	25.0	25.5	22.5	26.0	32.0	19.0	30.5	27.0	23.5	26.0	29.5	30.0	23.5	**35.0**	34.5
ER	20.0	21.0	28.0	37.0	36.5	9.9	30.5	26.5	19.5	20.5	26.0	21.0	23.5	40.5	**41.5**
CI	30.9	37.0	32.0	31.0	36.5	3.0	38.0	42.0	19.0	29.5	35.5	37.0	36.0	**65.5**	58.5
Avg	27.3	29.4	31.7	31.4	33.5	18.8	32.5	36.0	26.8	29.7	32.7	34.1	35.5	51.1	**54.9**

aligns with practical scenarios and involves fewer frames than most video LLMs. For VideoChatGPT-Bench and the video QA benchmark, which may involve videos spanning several minutes, we utilize the global-local input approach. In this configuration, the global frame number is set to 64. All videos are resized into 224*224. We train ST-LLM using a learning rate of 2e-5 and a batch size of 128 for 2 epochs.

Training Datasets. Citing findings from Jin et al. [20], the sequential process of first conducting image instruction tuning, followed by video instruction tuning and image-video joint pretraining, had a negligible impact on video dialogue performance. Hence, in light of InstructBLIP's prior training on extensive image data, we exclusively focus on video instruction tuning. Following [25], we leverage video instruction data sourced from a variety of datasets, including VideoChatGPT-100k [33], VideoChat-11k [24], Webvid [5], NExT-QA [47], CLEVRER [52], Kinetics-710 [21], and Something-Something-2 [17]. All samples are uniformly formatted in accordance with [25]. More details about the

Table 2: Comparisons on video-based generative performance benchmarking.

Method	Correct	Detail	Context	Temporal	Consist	Mean Score
VideoLLaMA [53]	1.96	2.18	2.16	1.82	1.79	1.98
LLaMA-Adapter [54]	2.03	2.32	2.30	1.98	2.15	2.16
VideoChat [24]	2.23	2.50	2.53	1.94	2.24	2.29
VideoChatGPT [33]	2.40	2.52	2.62	1.98	2.37	2.38
BT-Adapter [31]	2.68	2.69	3.27	2.34	2.46	2.69
VTimeLLM [19]	2.78	**3.10**	3.40	2.49	2.47	2.85
Chat-UniVi [20]	2.89	2.91	3.46	2.89	**2.81**	2.99
LLaMA-VID-7B [27]	2.96	3.00	3.53	2.46	2.51	2.89
LLaMA-VID-13B [27]	3.07	3.05	3.60	2.58	2.63	2.99
VideoChat2 [25]	3.02	2.88	3.51	2.66	**2.81**	2.98
ST-LLM (Ours)	**3.23**	3.05	**3.74**	**2.93**	**2.81**	**3.15**

dataset and the organization of instructions can be found in the appendix. An experiment on joint image-video training is also presented in the appendix.

4.2 Quantitative Result

MVBench Performance. The evaluation results on MVBench are presented in Table 1. The evaluation metrics included Action Sequence (AS), Action Prediction (AP), Action Antonym (AA), Fine-grained Action (FA), Unexpected Action (UA), Object Existence (OE), Object Interaction (OI), Object Shuffle (OS), Moving Direction (MD), Action Localization (AL), Scene Transition (ST), Action Count (AC), Moving Count (MC), Moving Attribute (MA), State Change (SC), Fine-grained Pose (FP), Character Order (CO), Egocentric Navigation (EN), Episodic Reasoning (ER), Counterfactual Inference (CI), and the average of all 20 metrics (Avg). It is evident that on such a challenging benchmark, the performance of most multimodal LLMs falls significantly short of satisfactory levels, even barely surpassing random performance by less than 10%. Our only formidable competitor is VideoChat2 [25]. Through direct substitution of CLIP with a dedicated video encoder, coupled with a three-stage training process and comprehensive instruction tuning for the visual encoder, Q-Former, and LLM, VideoChat2 has yielded results that significantly surpass those of previous models. In contrast, we solely carried out single-stage video-only instruction tuning while maintaining the weights of CLIP and Q-Former frozen. Even under these conditions, ST-LLM still attained the top performance, with an average score surpassing VideoChat2 by a notable margin of 3.8%. Specifically, ST-LLM showcased leading results across all 11 tasks, exhibiting a particularly significant advantage in motion-related tasks where other models struggled. In three motion-related tasks, ST-LLM achieved an average score of 59.2%, far outperforming VideoChat2's 36.3%. However, ST-LLM encounters challenges in fine-grained tasks, such as Fine-grained Action (FA) and Fine-grained Pose (FP). This could be attributed to CLIP's limitations as an image encoder, which may struggle to capture low-level spatial-temporal features required for these tasks.

Table 3: Comparisons on zero-shot question-answering, including MSVD-QA [46], MSRVTT-QA [48], and ActivityNet-QA [10].

Method	LLM	MSVD-QA		MSRVTT-QA		ActivityNet-QA	
		Acc	Score	Acc	Score	Acc	Score
FrozenBiLM [50]	DeBERTa-V2	32.2	-	16.8	-	24.7	-
VideoLLaMA [53]	Vicuna-7B	51.6	2.5	29.6	1.8	12.4	1.1
LLaMA-Adapter [54]	LLaMA-7B	54.9	3.1	43.8	2.7	34.2	2.7
VideoChat [24]	Vicuna-7B	56.3	2.8	45.0	2.5	26.5	2.2
VideoChatGPT [33]	Vicuna-7B	64.9	3.3	49.3	2.8	35.2	2.7
BT-Adapter [31]	Vicuna-7B	67.5	3.7	57.0	3.2	45.7	3.2
Chat-UniVi [20]	Vicuna-7B	65.0	3.6	54.6	3.1	45.8	3.2
LLaMA-VID [27]	Vicuna-7B	69.7	3.7	57.7	3.2	47.4	3.3
LLaMA-VID [27]	Vicuna-13B	70.0	3.7	58.9	3.3	47.5	3.3
VideoChat2 [25]	Vicuna-7B	70.0	**3.9**	54.1	3.3	49.1	3.3
ST-LLM (Ours)	Vicuna-7B	**74.6**	**3.9**	**63.2**	**3.4**	**50.9**	3.3

VideoChatGPT-Bench Performance. VideoChatGPT-Bench, also referred to as the Video-Based Generative Performance Benchmark, primarily evaluates the capability of video conversation. Utilizing GPT-3.5 [34], it assigns a score to the generated content across five aspects: Correctness of Information (Correct), Detail Orientation (Detail), Contextual Understanding (Context), Temporal Understanding (Temporal), and Consistency (Consist). As depicted in Table 2, the recently proposed state-of-the-art models [20,25,27] exhibit similar performance levels on this benchmark. However, due to the inherent instability of GPT evaluation, there is considerable variation in the performance of each model across different tasks. Despite this variability, ST-LLM consistently performs admirably across all tasks, while demonstrating significant advantages in terms of the mean score compared to previous models.

Zero-shot Video-question Answering Performance. In Table 3, we present the zero-shot video question-answering performance on several commonly used video-text datasets, including MSVD-QA [46], MSRVTT-QA [48], and ActivityNet-QA [10]. Following the methodology outlined in [33], we utilize GPT-3.5 to evaluate the accuracy and score of the generated results. As is depicted, STLLM demonstrates significant advantages on MSVD and MSRVTT, surpassing the previous SOTAs by 4.6% and 4.3% respectively. However, STLLM exhibits a slightly smaller advantage compared to VideoChat2 on the ActivityNet. This could be attributed to the requirement for a potent video encoder for this dataset.

4.3 Ablations and Analysis

Ways of Inputting video tokens. The core of this work lies in modeling joint spatial-temporal video tokens using LLM. Table 4 presents the results of different methods of inputting video tokens. As depicted, without video tuning, the approach of joint spatial-temporal input is considerably less effective compared to directly averaging the input over the temporal dimension. This discrepancy might stem from inconsistencies between training and testing. However, after

Table 4: The ablation study on the methods of inputting video tokens to LLMs. The baseline configurations include no instruction tuning and mean pooling input.

Method	Avg on MVBench
Instructblip Baseline	
Mean Pooling	42.0
S-T Tokens in LLM	35.5
Video Instruction Tuning	
Mean Pooling	47.8
S-T Tokens in LLM	53.8
+Masking & MVM Loss	**54.9**

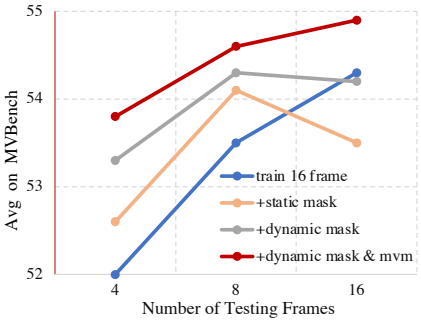


Fig. 4: The ablation study on the effect of dynamic masking and MVM loss.

Table 5: The ablation study on global-local input module. The "Local" and "Global" mean the joint temporal-temporal sequence and average pooling sequence respectively.

Method	Avg on MVBench	Video-ChatGPT Bench				
		Correct	Detail	Context	Temporal	Consist
Global Only	48.3	3.01	2.92	3.58	2.61	2.61
Local Only	**54.9**	3.08	2.99	3.63	2.75	2.69
Local+Global (simply add)	50.1	3.07	2.88	3.58	2.59	2.50
Local+Global (adapter)	54.7	**3.23**	**3.05**	**3.74**	**2.93**	**2.81**

training, the effectiveness of the joint spatial-temporal input significantly surpasses that of the mean pooling input, indicating that LLMs can effectively model spatial-temporal sequences, and directly inputting all tokens is a superior approach compared to compression before input. Furthermore, incorporating dynamic masking and the MVM loss based on the joint spatial-temporal sequence has further improved the performance of our model.

Dynamic Masking Strategy and MVM Loss. In Fig. 4, we conduct a more detailed ablation study on the effect of dynamic masking and MVM loss. It is evident that when the training frame count remains fixed, the performance of inputting all video tokens is not very robust. Particularly, when there is a discrepancy between the training and testing frame, the performance noticeably decreases. Meanwhile, when we randomly mask video tokens at a static 50% rate, this issue persists, with effectiveness akin to training with a fixed 8-frame setup. However, employing dynamic masking shows some improvement in overall performance, and notably, the robustness to varying testing frame counts is significantly enhanced. Furthermore, the inclusion of the MVM loss built upon the masking further enhances performance, with greater improvements observed as the sequence length increases.

Global-Local Input Module. The ablation results on the global-local module are presented in Table 5, where we compare the performance of global features extracted from more frames and local features from fewer frames, as well as their combination. As shown, on MVBench, the performance of global features

Table 6: The ablation study on the basic image models and the fine-tuning methodologies of LLM. We report the Avg on MVBench.

Baseline Model	LLM	LLM Parameters	Meanpooling Baseline	ST-LLM Full Model
MiniGPT-4 [55]	Vicuna-7B v0	LoRA	40.9	45.6
		Full Fine-tuning	46.4	50.5
InstructBLIP [14]	Vicuna-7B v1.1	LoRA	45.2	51.9
		Full Fine-tuning	47.8	54.9

Table 7: The analysis on the designs of inputting S-T video tokens into LLM.

Method	Avg on MVBench
S-T Tokens in LLM (baseline)	**54.9**
+frame-frame separators	54.0
+video-text separators	54.8
+S-T position embeddings	52.2

Table 8: The analysis on the designs of dynamic masking.

Method	Avg on MVBench
w/o masking (baseline)	53.8
$\rho \sim \mathbb{U}(0.3, 0.7)$	54.2
$\rho \sim \mathbb{N}(0.5, 0.2)$	54.3
$\rho \sim \mathbb{N}(0.5, 0.1)$	**54.9**

is significantly inferior to the joint S-T features, so combining global and local features did not lead to improvement. This may be due to the fact that MVBench mainly consists of short videos lasting a few to a dozen seconds, where a large number of frames might not be necessary. However, on VideoChatGPT Bench, where videos are parsed at the minute level, the discrepancy between global and local features has notably diminished, with their amalgamation resulting in a discernible enhancement. This underscores the efficacy of the Global-Local Input module in processing extended video content. Moreover, the direct incorporation of global and local features resulted in a performance decline, presumably due to its interference with the representation learning of the LLM.

Basic Image Models & LLMs. In Table 6, we conduct ablations on both the basic image chatbots and the LLM fine-tuning strategy. As depicted in the table, regardless of whether it's LoRA [18] tuning or full fine-tuning, or different basic image models, ST-LLM demonstrates notable enhancements compared to the mean pooling baseline. This suggests that our approach can be adeptly applied across diverse scenarios, such as employing more resource-efficient LoRA tuning on larger models, or transferring to more potent image-dialogue models.

Designs of ST-LLM. In Table 7 and Table 8, we investigated several alternative design possibilities for ST-LLM. Primarily, we explored various separator designs. It's noteworthy that almost all cross-modal LLMs incorporate separator designs, with some models even integrating separators between multiple images or frames [2, 4]. However, we discovered that when employing LLMs to model spatial-temporal sequences, these separators are superfluous. Subsequently, we address positional encodings. Numerous approaches integrate temporal positional encodings before inputting video tokens into LLMs [32, 33, 53]. However, in our methodology, we observed that incorporating absolute positional encodings to all visual tokens significantly impairs performance. Lastly, we consider the design of dynamic masking. Our findings suggest that all distributions

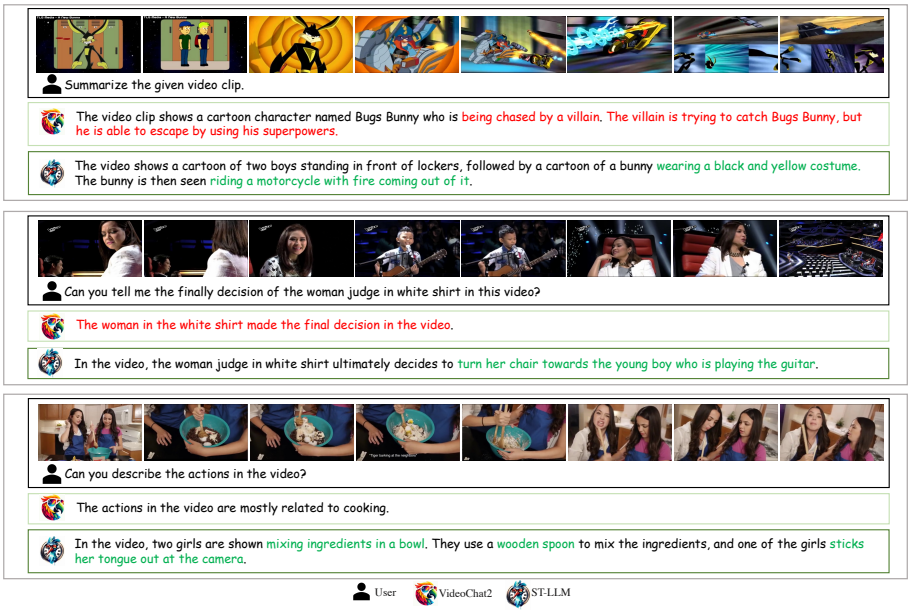


Fig. 5: Qualitative results from real-world videos.

of masking rates can yield some improvement, yet distributions with smaller standard deviations exhibit superior performance.

4.4 Qualitative Results

In Fig. 5, we present a qualitative comparison. As illustrated, ST-LLM excels in adhering to instructions and delivering precise responses. More importantly, ST-LLM showcases superior sensitivity to temporal sequences and actions, thereby validating the effectiveness of our modeling approach. For further quantitative results and failure cases, please refer to the appendix.

5 Conclusion

In this paper, we present ST-LLM, a straightforward yet robust video large language model. Our aim is to achieve effective video comprehension by leveraging LLM to model video tokens, marking the inception of a joint spatial-temporal-text modeling paradigm. Additionally, we introduce a dynamic masking strategy and a global-local input module to enhance this framework. Remarkably, ST-LLM excels in encoding and comprehending spatial-temporal sequences while addressing concerns related to efficiency, stability, and modeling lengthy videos, all with reduced training resource requirements. Through extensive experimental analysis, we demonstrate the effectiveness of ST-LLM and its internal design, resulting in state-of-the-art performance across multiple video LLM benchmarks.

Acknowledgements. This work was supported by National Natural Science Foundation of China (No. 62172021). We are also grateful to the support from Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology.

*: Equal Contribution, work done during internship at ARC Lab, Tencent PCG.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022)
3. Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al.: Palm 2 technical report. arXiv preprint arXiv:2305.10403 (2023)
4. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv preprint arXiv:2308.01390 (2023)
5. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1728–1738 (2021)
6. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
7. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. vol. 2, p. 4 (2021)
8. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators
9. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
10. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 961–970 (2015)
11. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
12. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)
13. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) (2023)

14. Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: towards general-purpose vision-language models with instruction tuning. arxiv. Preprint posted online on June **15**, 2023 (2023)
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
16. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
17. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE international conference on computer vision. pp. 5842–5850 (2017)
18. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021)
19. Huang, B., Wang, X., Chen, H., Song, Z., Zhu, W.: Vtimellm: Empower llm to grasp video moments. arXiv preprint arXiv:2311.18445 (2023)
20. Jin, P., Takanobu, R., Zhang, C., Cao, X., Yuan, L.: Chat-univi: Unified visual representation empowers large language models with image and video understanding. arXiv preprint arXiv:2311.08046 (2023)
21. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
22. Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., Liu, Z.: Otter: A multi-modal model with in-context instruction tuning. arXiv preprint arXiv:2305.03726 (2023)
23. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
24. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: Chat-centric video understanding. arXiv preprint arXiv:2305.06355 (2023)
25. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: Mvbench: A comprehensive multi-modal video understanding benchmark. arXiv preprint arXiv:2311.17005 (2023)
26. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked teacher: Towards training-efficient video foundation models. arXiv preprint arXiv:2303.16058 (2023)
27. Li, Y., Wang, C., Jia, J.: Llama-vid: An image is worth 2 tokens in large language models. arXiv preprint arXiv:2311.17043 (2023)
28. Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122 (2023)
29. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
30. Liu, R., Huang, J., Li, G., Feng, J., Wu, X., Li, T.H.: Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6555–6564 (2023)
31. Liu, R., Li, C., Ge, Y., Shan, Y., Li, T.H., Li, G.: One for all: Video conversation is feasible without video instruction tuning. arXiv preprint arXiv:2309.15785 (2023)

32. Luo, R., Zhao, Z., Yang, M., Dong, J., Qiu, M., Lu, P., Wang, T., Wei, Z.: Valley: Video assistant with large language model enhanced ability. arXiv preprint arXiv:2306.07207 (2023)
33. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv preprint arXiv:2306.05424 (2023)
34. OpenAI: Chatgpt. https://openai.com/blog/chatgpt (2023)
35. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)
36. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
37. Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F.: Beit v2: Masked image modeling with vector-quantized visual tokenizers. arXiv preprint arXiv:2208.06366 (2022)
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
39. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
40. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)
41. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model (2023)
42. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022)
43. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
44. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
45. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
46. Wu, Z., Yao, T., Fu, Y., Jiang, Y.G.: Deep learning for video classification and captioning. In: Frontiers of multimedia research, pp. 3–29. ACM (2017)
47. Xiao, J., Shang, X., Yao, A., Chua, T.S.: Next-qa: Next phase of question-answering to explaining temporal actions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9777–9786 (2021)
48. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5288–5296 (2016)
49. Xue, H., Sun, Y., Liu, B., Fu, J., Song, R., Li, H., Luo, J.: Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. arXiv preprint arXiv:2209.06430 (2022)

50. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Zero-shot video question answering via frozen bidirectional language models. Advances in Neural Information Processing Systems **35**, 124–141 (2022)
51. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv preprint arXiv:2304.14178 (2023)
52. Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., Tenenbaum, J.B.: Clevrer: Collision events for video representation and reasoning. arXiv preprint arXiv:1910.01442 (2019)
53. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023)
54. Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., Qiao, Y.: Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv preprint arXiv:2303.16199 (2023)
55. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)